Automated Multi-Robot Assembly of Compliance Optimized Structures

Yicong Fu
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332
yfu97@gatech.edu

James Todd Gloyd
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332
jgloyd3@gatech.edu

Irina Kostitsyna
KBR at NASA Ames Research Center
Moffett Field, CA 94043
irina.kostitsyna@nasa.gov

***Abstract*—Autonomous assembly of large structures is one of the fundamental challenges on the way towards NASA's objectives of deep space exploration. In this work, we propose an algorithmic framework to optimize the assembly process of a prescribed target structure by constraining the assembly effort as well as maintaining structural soundness throughout the process. This framework uses structural topology optimization with assembly effort metrics to generate checkpoints for robotic traversal algorithms. Assembly effort is quantified by the Wasserstein metric between consecutive structural configurations during the assembly process. The robotic assembly task is split into two subtasks, where we first optimize for a set of key frames, then perform reconfiguration between consecutive frames. Key frames are optimized by adopting topology optimization techniques to reduce assembly effort and maintain structural integrity during the assembly process, while reconfiguration between key frames is performed using a path planning algorithm with a minimum weight maximum matching approach on a bipartite graph. We employ a Crystalline robot model in which each structural element is capable of locomotion through the structure and locking into place with neighboring elements after reaching its destination. An example assembly of a two-dimensional cantilever beam under volume constraints and structural compliance considerations is presented to demonstrate the approach. Finally, we conclude by discussing possible future extensions to this work, including adoption of better metrics, extension to three-dimensional large-scale problems, and exacting finer control of structural integrity during the path-planning phase.**

TABLE OF CONTENTS

1. INTRODUCTION

Large extraterrestrial structures will assuredly require autonomous assembly methods to construct [1–4]. Unlike manual construction via extra-vehicular activity, assembly by robotic swarms allows for robust, scalable, and time-flexible construction in hazardous environments. Such multi-robot systems can consist of rather simple robots; nevertheless, from combination of simple actions of individual robots, complex behavior of the system as a whole can emerge. Realizing such complex emergent behavior, however, is a challenging algorithmic task [5–7]. Not only must the robotic assembly system ensure the final structure complies with the specification requirements, but also that the intermediate states of the construction satisfy a number of physical constraints. Looking towards these needs, this paper presents an algorithmic framework for robotic construction of discrete, on-site assembly of structures that

1. produces a sequence of intermediate structure states converging to a targeted solution; each of the intermediate structures is optimized for sustaining the weight of the robotic system, and
2. directs the autonomous reconfiguration of the structure following the sequence of intermediate states by the robots, with each step being optimized to minimize the total reconfiguration timespan.

The work presented here addresses some unique considerations necessary for automated multi-robot assembly. Foremost among those challenges is the requirement of continuous structural integrity throughout the assembly process, while limited in the amount of material available. Consideration must also be made regarding the time and effort required by the assembly process, especially in extraterrestrial applications where time and energy spent on assembly may be at a premium. To address these challenges, we approach the problem by combining topology optimization with robotic traversal algorithms. Topology optimization is used to generate a sequence of intermediate structural states which are guaranteed to meet or exceed certain levels of structural integrity.

Topology optimization, in structural settings, applies optimization methods to a parameterization of a structure's shape in order to optimize various structural objectives. Because of the non-standard shapes produced by topology optimization, in most applications, the resulting structure is assumed to use additive manufacturing or casting for production. However, additive manufacturing and casting are limited in build volume by the equipment used and not viable for large, in-space fabrication. Therefore, in-space construction of a shape-optimized structure requires a radically different construction method. To this end, we propose a multi-robot assembly framework, which, if approached naively, could result in structural failure at an intermediate step. We ensure integrity of the intermediate structure by incorporating new constraints into the shape optimization formulation. Robotic path planning algorithms are then used to determine the most efficient sequence and set of paths to reconfigure from one structural state to the next, essentially stringing the structural states together. Results from this approach are presented for the assembly of a compliance optimized, cantilevered beam.

This work focuses specifically on the problem of assembly in a gravity field, where each structural component is also a robot. That is to say, each cellular element of the structure is capable of robotic locomotion, and also capable of locking together with other elements and transferring structural loads when stationary. However, these specifics are not critical to the overall approach. For example, a similar approach to the one described here can be applied to on-orbit situations, where natural frequencies of the structural system may become more important constraints than the stress levels. A similar approach could also be used for alternative robot systems, for example systems where the robotic elements are separate from the structural components, and in such alternate systems, the number of robots involved in the assembly may change the optimization result, though the process used to arrive at that result would remain the same.

2. Topology Optimization

First introduced by Bendsøe and Kikuchi [8] in 1988, topology optimization has become a promising research area and has seen widespread industrial applications in aerospace, mechanical design, construction, and automotive engineering [9–13]. This work is focused on structural topology optimization, which is a method of generatively designing potentially complex structures based on a predefined domain and a set of boundary condition, without needing any a priori knowledge regarding the general form of such structures. Topology optimization offers the capability to design efficient, light-weight structure configurations, and is especially beneficial over traditional design approaches when weight is a critical factor, such as for structures in aerospace applications. Facilitated by advancements of high-performance computers and scalable numerical algorithms, topology optimization has become applicable to large-scale problems, including static problem [14, 15], eigenvalue problem [16–18] and nonlinear physics [19]. Furthermore, topology optimization can be used to design optimized configurations that are governed by physics other than static elasticity, including fluid mechanics, heat transfer, acoustics, electromagnetism, and optics to name a few.

3. Robotic Framework

In this work, we assume a robot model analogous to the Crystalline robot system [20]. The functional simplicity of crystalline robots could potentially pave the way for the realization of physical robots, that would possess adequate structural properties to serve as a foundation for our architecture.

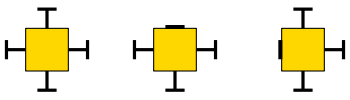


Figure 1: A 2D Crystalline robot with four arms extended (left), and with one arm contracted (middle and right).

In the 2D Crystalline robot system, square robots are arranged on a square lattice (refer to Figures 1 and 2). Each robot has four extendable arms, with which it can grip on to its neighbors, where each arm can extend half a unit distance in length. By contracting and extending their arms, robots can pull and push on their neighbors, thus resulting in reconfiguration of the structure. In this work we assume a constant strength formulation of the Crystalline robot model [21].

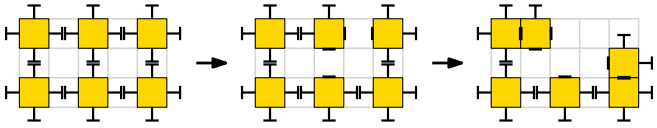


Figure 2: Example of reconfiguration by 2D crystalline robots: initially six robots are attached to each other forming a 2×3 grid, all their arms are extended (left); two robots in the top row disconnect from some of their neighbors by detaching and contracting their arms (middle); two pairs of robots contract their arms, which leads to two robots moving by one unit of distance (right).

More specifically, we assume that any contracting or extending arm cannot move more than two crystalline robots at a time. Figure 2 shows a valid reconfiguration step. Note that it is assumed that in any terminal, i.e. stationary, configuration, all arms of the crystalline robots must be extended.

It is not always the case that a crystalline structure can reconfigure between two valid states. For example, as the robots can only perform push and pull operations, a straight-line chain of robots cannot reconfigure in any state with depth or height of the structure higher than one. Aloupis et al. [21] have proven that, in 2D, given an initial configuration of crystalline robots S and a target configuration T, there always exists a reconfiguration strategy of $O(n)$ synchronized moves, as long as both S and T can be partitioned into blocks of 2×2. Their approach was to consider 2×2 metamodules as atomic building blocks. Note that, in a terminal configuration, a 2×2 metamodule occupies a 4×4 square (shifted by half a unit along both axes). Thus, in an intermediate configuration, two metamodules can occupy the same 4×4 square if some of their arms are contracted. This gives enough flexibility to the system to allow the metamodules to traverse through the interior of the structure.

In this paper we assume each element of our optimized structure is one such 2×2 metamodule of four crystalline robots. When planning a schedule for the robots corresponding to the reconfiguration between a pair of topologically optimized frames, we model the state of the system as a grid of metamodules (see Figure 3). Each cell in our grid can have a weight of zero, one, or two, corresponding to the number of metamodules occupying that cell. Applying the results of Aloupis et al., we assume a unit of weight can be shifted between neighbouring cells in a constant number of steps. Thus, given a pair of consecutive keyframes x_i and x_{i+1}, we identify the cells $x_i \setminus x_{i+1}$ that are occupied in x_i but are not occupied in x_{i+1}, and move the corresponding metamodules to the locations of $x_{i+1} \setminus x_i$, along shortest paths through the interior of the structure (refer to Figure 10 and Section 7 for more details).

This robotic model predicates translational motion along the coordinate axes. Because of this, paths through the structure are measured using L^1 distance (further discussed in Section 5) since that measure is consistent with such a model. Extending this to 3D motion along the coordinate axes is straightforward, but if a different robotic model is use where other types of motion like rotations are possible, then a different measure and appropriate path planning would likely be more suitable. However, making such changes to the path planning and other aspects could still be integrated with the rest of the presented approach.

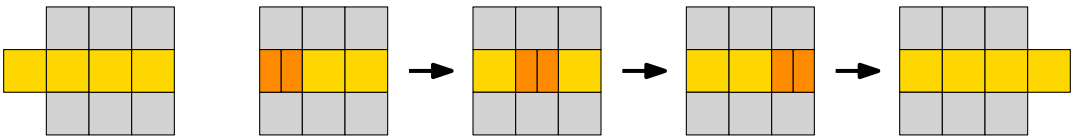


Figure 3: An example of a reconfiguration assuming 2×2 metamodules: Yellow cells depict metamodules participating in the reconfiguration. Orange cells contain two metamodules (eight crystalline robots).

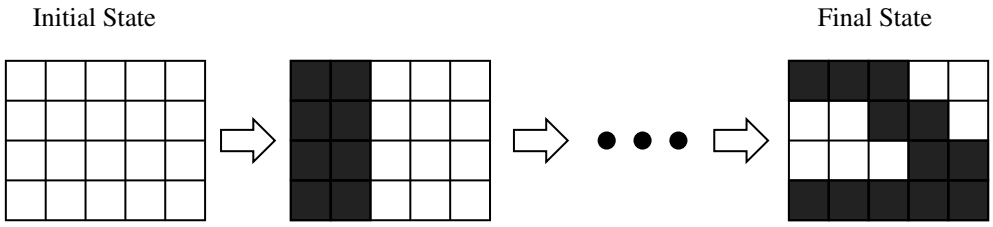


Figure 4: A simple assembly sequence visualization showing an empty domain as the initial state and a rudimentary cantilevered design as the final state.

4. Problem Depiction

For this problem, we seek an assembly sequence which leads to a compliance optimized final design, while also limiting the compliance of the intermediate states under self-weight within a gravity (body force) field. A compliance optimized cantilever beam is a classic result in topology optimization, and compliance optimization with the addition of self weight is also well investigated. By limiting the compliance of the intermediate stages during assembly, we ensure the structure will not collapse during the assembly process. Furthermore, we are able to minimize or otherwise control the total assembly time.

The assembly sequence itself is represented by a set of design variables which indicate where material should be, within the domain, at particular times in the assembly process. Once the assembly sequence is determined, the robotic path to get from one intermediate state to the next is a much easier problem to solve. However, finding an assembly sequence which is time efficient, robotically achievable, and satisfies structural constraints is a non-trivial task.

We go about solving this problem starting with two fixed endpoints: an initial design (the unassembled state) and a compliance optimized design (the final state). The compliance optimized design is chosen for the final state in this work since it is a classic and well studied result.

We then define some number of intermediate states between these endpoints, which we can think of like a deck of cards which show the assembly sequence as it progresses from the initial state to the final structure. A simple illustration of this conceptualization can be seen in Figure 4. Now the problem becomes one of determining the frames between these two endpoints, i.e. the intermediate design states, which the robots can achieve in the smallest amount of time while maintaining structural integrity. In addition to the satisfaction of the structural constraints, there are two things happening in these intermediate frames. First, the structural material is being introduced to the domain, i.e. brought in from some depot. Second, the material is being moved into position to form the goal structure, that is the cantilever beam.

Each frame of the assembly sequence is divided into a grid of cells, and each cell has an associated design variable which indicates the presence or absence of material at the cell's location at that particular point in the assembly.

5. Lower Bound on Reconfiguration Cost

For this problem, we use a measure of the difference between two distributions, referred to as the Wasserstein metric, to provide a lower bound on the cost of a given reconfiguration. This measure of difference between distributions is the minimum total movement of material required to transform one distribution to another, based on costs associated with moving material between locations in the distributions. As such, between two frames of the design sequence, the Wasserstein metric measures the best-case scenario distance which must be traversed by the robots to move between those two frames (while ignoring all requirements on robot motion and structural integrity, including the connectivity requirement). The true total distance traversed by the robots between frames will be larger than the Wasserstein metric since the robots operate under the additional constraint of moving through or adjacent to the existing structure, which can exclude some of the more direct paths. We use the Wasserstein metric in our estimator because doing so allows us to quickly, and differentiably, measure the cost of an assembly sequence without computing the entire multi-robot path. Moreover, the Wasserstein metric is valid for both discrete and continuous values of the design variables. This is important because even though the goal of this work is to obtain discrete designs, we are using a continuous formulation to tackle what would otherwise be an intractable problem. The discrete problem is equivalent to an integer programming problem, which is NP-complete, whereas the continuous version can be addressed with polynomial time optimization algorithms.

For a given assembly sequence, we define a lower bound on the cost of that sequence as the sum of the Wasserstein metric between each consecutive frame. The Wasserstein metric measures the minimum movement of material necessary to transform one frame into the next. This value is computed by finding the minimum cost paths for all material which needs to be moved to transform one distribution into another, given the cost of moving material between any two positions. Evaluating this distance requires solving a linear program optimization, as discussed below.

First though, we note use of the L^1 norm between two points to represent the distance between cell locations in this calculation because this measure of nodal distance corresponds to the type of motion inherent in the robotic model used (as described in Section 3).

An illustration for a toy version of this sub-problem is shown in Figure 5, where there are two sources of material, i and j, and two destinations, k and l. The optimal transport for

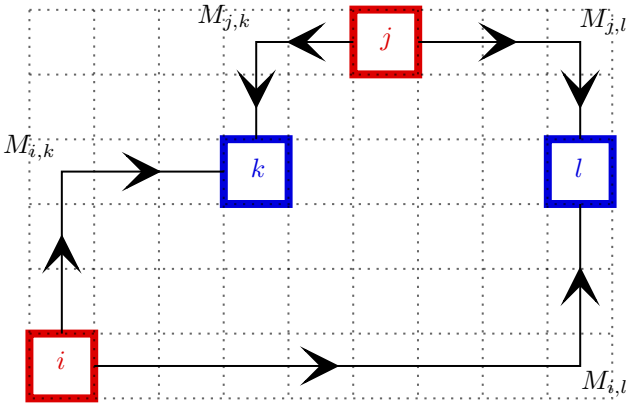


Figure 5: An illustration of optimal transport, Wasserstein metric showing two sources, i and j, two sinks, k and l, and four L^1 paths by which the material can move from the sources to the sinks. In this example, these paths have lengths $M_{i,k} = 6$, $M_{i,l} = 11$, $M_{j,k} = 4$, and $M_{j,l} = 5$.

such a system consists of solving the linear programming optimization problem shown below in (1).

$$\begin{aligned} \min_{\gamma \in \mathbb{R}^{\eta \times \eta}} \quad & \sum_{i,j} \gamma_{i,j} M_{i,j} \,, \\ \text{s.t.} \quad & \gamma \cdot 1 = a \,, \\ & 1 \cdot \gamma = b \,, \\ & \gamma \geq 0 \,. \end{aligned} \tag{1}$$

In the above optimization problem, the M matrix contains the distance between any two nodes in the domain, the a and b vectors represent the proportional material distribution among the nodes in the initial configuration and final configuration respectively, η is the number of nodes (i.e. positions where material can be placed) in the system, and $\gamma_{i,j}$ is the proportion of material transported from node i to node j. For our specific application, since the robotic model used here is only able to travel using a combination of single unit vertical and horizontal steps, the M matrix here contains the L^1 distances between the various component locations in the system. The a and b vectors are determined by the distribution of material in two subsequent states of the assembly sequence.

The problem shown in (1) is fairly well studied, which is part of the motivation to use the Wasserstein metric in this work. In order to solve this optimal transport problem and compute the Wasserstein metric, we use [22], as well as the toolbox involved in that work. As described in the beginning of this section, this computation gives a lower bound which is then used as part of the overall topology optimization.

6. Optimization Framework

The overall topology optimization problem takes the form of a compliance minimization, over all configurations in the assembly sequence, subject to nonlinear inequality constraints on a measure of the assembly time, as estimated by the Wasserstein metric, and linear inequality constraints on the mass of each frame. The goal of this optimization setup is to generate a sequence of intermediate designs, which can be constructed by the robots, leading to the goal configuration. This includes the implicit assumption that the structure does not collapse or otherwise fail during the assembly, leading to a tightly coupled problem.

We note here that originally, we chose a completely empty design as the initial state for the optimization, but we found that the optimization process had a tendency to break the task of assembly into two parts. First, in the early frames of the assembly sequence, the optimization would put the vast majority of the available material into the domain in one clump near the root. Next, in the later frames of the assembly, the optimization reconfigures the material into a sequence of structures approaching the final state. Due to this natural split in tasks, we decided to set the initial state for the assembly sequence as a solid block of material grouped at the root of the design domain. As such, the assembly sequence resulting from our optimization defines the reconfiguration of the structure into the final cantilevered beam design, from a cluster of material at the root.

Assembly Time Constraint

We want to minimize, or at least limit, the time needed for the assembly process, which is where the Wasserstein metric comes into play. As discussed in Section 5, the Wasserstein metric acts as a surrogate for robotic path planning: a measure of the amount of movement required by the robots to get from one frame in the design sequence to the next. To that end, this measure is incorporated into the topology optimization process to inform generation of the assembly sequence. Again, as discussed, the Wasserstein metric provides the lower bound on the amount of movement required by the robots for reconfiguration.

An important observation here is that the sum of the Wasserstein metric over all frames of the assembly must be greater than or equal to the Wasserstein metric between the first and last frame. This is a consequence of the triangle inequality, which holds for the formulation of the Wasserstein metric which we use for this work [23], although this inequality does not necessarily hold for the all forms of the Wasserstein metric. Because we know this lower bound on the total Wasserstein metric, we can use this bound to generate an inequality constraint which limits the search of the design space to assembly sequences remaining within a certain proportion of the optimal value. That is to say, we calculate the best possible Wasserstein metric and constrain our optimization to be within some percentage of the best case scenario.

An interesting, albeit frustrating, consequence of the fact that the Wasserstein metric is a lower bound on the assembly cost (reconfiguration time) is that the optimization process tends to obfuscate the details of reconfiguration by concentrating most of the reconfiguration in a small number of frames. That is to say, the optimization is unintentionally incentivized to generate assembly sequences with large changes in the structure in a single step. This is fairly undesirable behavior, since assembly sequences with large changes in the design don't give much information to the robotic algorithms and

the intermediate states between two such frames do not have the same strength of guarantees of structural integrity. In order to address this tendency, we use an exponentiated sum of the Wasserstein metric, that is, we raise each calculated Wasserstein metric to some power before taking the sum.

The idea behind this exponentiation stems from the fact that the p-norm acts as a smooth approximation to the maximum of a vector. If we explicitly constrain the maximum Wasserstein metric, or some smooth approximation of the maximum, instead of the sum of the Wasserstein metrics, then that successfully results in assembly sequences with close to equal amounts of reconfiguration with each step. However, doing so does not penalize superfluous reconfiguration of all but the most expensive step, which is not desirable since it results in a larger total cost of assembly. Instead, we balance these goals by using a modest exponent, between two and six, which focuses on reducing the largest value of the Wasserstein metric over the entire assembly sequence, but still penalizes any unnecessary reconfiguration between less active frames. Selection of this exponent is explored further with numerical experiments in Section 7.

Furthermore, we are still able to establish a minimum lower bound for this exponentiated sum,

$$\sum_{i=1}^{n}(w_i)^p \geq \frac{W^p}{n^p}n\,. \tag{2}$$

Here n is the number of steps in the assembly, w_i is the Wasserstein metric between frame $i-1$ and frame i, W is the Wasserstein metric between the initial state and the final state, and p is any positive exponent. This bound follows from the Hölder's Inequality, and, as discussed earlier, we use this bound to generate a constraint for our optimization to be within a certain percentage of the best case scenario.

Mass Constraint

In addition to constraining the Wasserstein metric, we also place linear constraints on the amount of material in the frames. Specifically, we constrain the difference in amount of material between frames of the assembly sequence to be greater than some lower bound, that is the sum of material in frame $i+1$ must be greater than or equal to the sum of material in frame i plus the prescribed lower bound. For example, if we set the lower bounds to be zero, then this constrains the amount of material to be non-decreasing as the assembly progresses. If instead we set the lower bounds to be some positive number, then this constrains the assembly sequence to be strictly increasing in material between frames by at least the given amount. Note that this constraint is not entirely necessary since removing and adding material unnecessarily increases the Wasserstein metric as discussed earlier and would therefore be penalized, but our tests indicate that including this constraint leads to improved results, which is likely due to the additional information regarding the nature of the problem conveyed to the optimization process through the constraint.

Compliance Objective

As mentioned many times throughout this paper, one of the critical components of this problem is the fact that structural integrity must be maintained during the entire assembly process. To account for this aspect of the problem during the topology optimization, we chose the objective of our optimization to be a minimization of the sum of compliance over each frame in the assembly sequence. Compliance of the structure is a measure of the deformation of all components, weighted by the load the component experiences. Consequently, minimizing compliance ensures that deformations are kept relatively small. Furthermore, compliance is a well studied and readily implemented metric in topology optimization.

Use of compliance minimization as our objective also protects the optimization from producing disjoint structures. This is because when a structural component under self weight, i.e. with body forces, becomes disconnected from all fixed boundaries, the compliance of that component essentially becomes unbounded since the body force will cause uncontrolled displacement. Since this is such a severe outcome, the any tendency to create disconnected structures is thoroughly suppressed.

Discretization Term

The topology optimization in this work takes place over a continuous design domain, but the real structure with which we are working is necessarily discrete. As mentioned above, solving the discrete version of this problem is computationally intractable, and therefore we use a continuous analog. However, we must remain somewhat faithful to the discrete problem we plan to map our result back to.

As such, following [24], we employ a discreteness metric, and include it as an additional term in the objective function which penalizes designs which are very far from being discrete, but still allows movement within the continuous domain. Specifically, the discretization term, $d(x)$, augmenting the objective is given by the vector function shown in (3).

$$d(x) = \frac{4x\cdot(1-x)}{m} \tag{3}$$

Where x are the design variables, m is the total number of design variables needed to specify the assembly sequence; division by m normalizes the function. This function is at a maximum when all of the design variables, x, are equal to one half, that is when they are furthest from their discrete values. As such, augmenting the compliance objective in this way results in a minimization which penalizes non-discrete values depending on a measure of how far the design is from being discrete. Importantly though, this penalization is not so severe for any one design variable that it impedes the design from exploring the design space. Instead, the optimization is enticed to limit the number of design variables with significantly non-discrete values at any one time.

7. NUMERICAL EXPERIMENTS

In this section, we present the numerical results of an autonomous assembly problem for the cantilever structure. We use two-dimensional problems to demonstrate our proposed approach because of its affordable computational cost, but the presented method is not restricted to any two-dimensional assumptions. For the two-dimensional structural analysis, we adopt the plane stress assumption, which would not be necessary for three-dimensional problems. In higher-dimensional problems, as long as the structural analysis and/or underlying physics are properly implemented for the situation, no modification is needed to the rest of this approach.

Problem Formulation

The design domain of this cantilever structure is shown in Figure 6. Figure 6a illustrates the domain geometry as well

as boundary conditions. Figure 6b shows the structured mesh that we used to solve the underlying finite element problems. Figure 6c shows the the target design of the cantilever structure, obtained via compliance minimization topology optimization. The design domain for the topology optimization problem is a two-dimensional domain with length $2L$ and height L. A fully clamped boundary condition is applied to the left edge. A concentrated load is applied to the lower right corner. A compliance-minimized result is presented in Figure 6c.

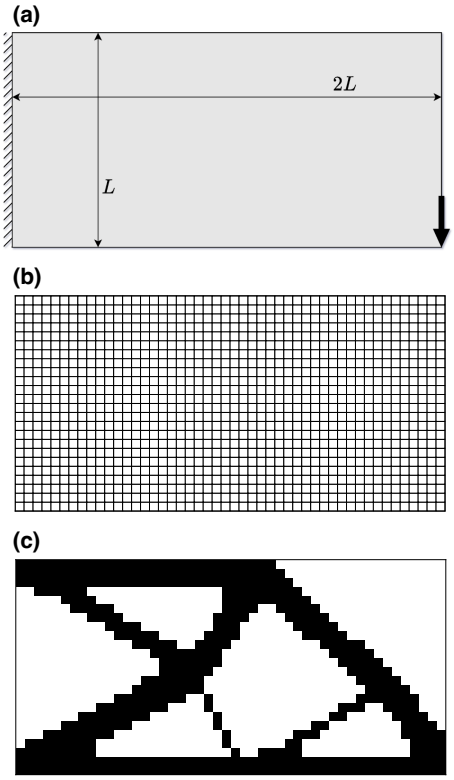


Figure 6: **(a)** Design domain, **(b)** finite element mesh (24×48 cells), **(c)** target design

To numerically solve the assembly problem, we propose a two-step sequential design strategy. First, we discretize the conceptually continuous assembly process by a set of key frames, and we solve a monolithic optimization problem that optimizes for all the key frames at once. Once we obtain all key frames, a robotic transfer strategy is employed by solving a robotic path planning problem that moves materials between two consecutive frames. These two steps will be further elaborated in the following subsections.

Optimization Framework

Design of the assembly process of such structure can be formulated as the following abstract optimization problem (4), as shown below.

$$\begin{aligned} \min \quad & \sum_{i=0}^{N} e(x_i, x_{i+1}) \,, \\ \text{w.r.t.} \quad & x_1, \dots x_N \,, \\ \text{subject to} \quad & \text{volume constraints} \,, \\ & \text{structural integrity constraints.} \end{aligned} \tag{4}$$

In (4), N is the number of key frames of the process that we seek to design. Design variables $x_1, \dots x_N$ represent the material distributions of all design frames. x_0 and x_{N+1} are the material distributions of the initial frame and target frame respectively, and remain constant within the scope of an optimization problem. e is a metric that quantifies or approximates the cost accrued by moving from one frame to another. In this work, we use the Wasserstein metric described in Section 5 to construct this metric. Lastly, volume constraints are constraints on the amount of material used for each key frame, and structural integrity constraints impose requirements such that the structures represented by key frames and intermediate states between key frames will not fail on self-weight. Also included in such constraint sets may be constraints that prevent the optimizer from generating structures disconnected from the clamped boundary condition, which cannot be physically assembled without auxiliary supports.

Optimization for Key Frames

In this work, the actual optimization problem we formulated takes a form that is slightly different from the general framework (4), which is shown below.

$$\begin{aligned} \min \quad & \max_{i=1,\dots,N} c(x_i) + \mu \sum_{i=1}^{N} d(x_i) \,, \\ \text{w.r.t.} \quad & x_1, \dots x_N \,, \\ \text{subject to} \quad & e(x_0, \dots, x_{N+1}) \le k e_{\text{ref}} \,, \\ & V(x_i) \le V(x_{i+1}) + \epsilon \quad \text{for } i = 0, \dots, N \,, \\ & x_0 = x_{\text{init}} \,, \\ & x_{N+1} = x_{\text{target}} \,. \end{aligned} \tag{5}$$

In (5), c in the objective is the compliance function of the structure under self-weight. d is the discreteness metric described in (3). For density-based topology optimization tasks, a well optimized topology is usually a discrete 0-1 design with d close to 0. Such discrete designs for regular problems such as volume-constrained compliance minimization problems can be achieved using material penalization such as SIMP [25] or RAMP [26]. For our problem, we add a regularization term as an extra guard to ensure the discreteness of the optimized design. μ is a regularization coefficient, which should be a large enough positive number to ensure effectiveness of the regularization, but not too large such that the contribution from compliance gets neglected. For the presented numerical study, the compliance part of the objective is normalized against the reference value so it will have magnitude $O(1)$. By definition, $d \in [0, 1]$, as can be seen by inspection of (3). As a result, we choose to use $\mu = 10$ such that the entire objective will have same order of magnitude as the compliance if d is small enough, and will be significantly larger than the compliance if d is large. Next, V is the volume function for the structural configuration. ϵ is a relaxation term, in this study, we use $\epsilon = 0.001 V_{\text{domain}}$, where V_{domain} is the volume of the entire mesh. Finally, e is a cost metric that measures the difference among all frames. k is a multiplier, and e_{ref} is the theoretical lower bound of such metric. In this study, a typical value of k is between 1.05 to 1.5.

There are different strategies for defining e, one example would be

$$e(x_0, \dots, x_{N+1}) = \sum_{i=0}^{N} w(x_i, x_{i+1}) \,, \tag{6}$$

where w is the Wasserstein metric described in Section 5. Such constraint function bounds the maximum cost of the entire process, and treats each step equally. Based on our preliminary study, one draw back of this approach is that such a constraint does not penalize drastic change between key frames. Suppose we have two candidate series of key frames $S_1 = \{x_1^{(1)}, x_2^{(1)}, \ldots, x_N^{(1)}\}$ and $S_2 = \{x_1^{(2)}, x_2^{(2)}, \ldots, x_N^{(2)}\}$. In the S_1 series, the structure is evolving gradually, while for the S_2 series, all frames are identical to the initial frame

$$x_0^{(2)} = x_1^{(2)} = x_2^{(2)} = \cdots = x_N^{(2)}.$$

In fact, for this case, we always have

$$e(S_1) \geq e(S_2) = w(x_0, x_{N+1}). \tag{7}$$

Because otherwise, we found a procedure to incrementally move from initial structure x_0 to final structure x_{N+1} with a total cost lower than the Wasserstein metric between x_0 and x_{N+1}, which contradicts the definition of the Wasserstein metric. Hence, results using (6) in the constraint will likely lead to series similar to S_2, which doesn't give enough information on the evolution process.

To circumvent such an issue, we need to constrain not only the total cost, but also penalize drastic changes between consecutive frames. To achieve this, we utilize the following exponentiated sum

$$e_p(x_0, \ldots, x_{N+1}) = \sum_{i=0}^{N} \left(w(x_i, x_{i+1})\right)^p , \tag{8}$$

where $p > 1$ is the penalization parameter. When $p = 1$ the exact Wasserstein metric is recovered. When $p > 1$, minimizing or constraining the metric e in equation (8) will penalize drastic changes between consecutive frames. It can be shown using Hölder's inequality that expression (8) has a lower bound, as demonstrated in (9).

$$\begin{aligned} e_p(x_0, \ldots, x_{N+1}) &\geq \frac{\left(\sum_{i=0}^{N} w(x_i, x_{i+1})\right)^p}{(N+1)^{p-1}} \\ &\geq \frac{\left(w(x_0, x_{N+1})\right)^p}{(N+1)^{p-1}} = e_{\text{ref}}. \end{aligned} \tag{9}$$

In practice, a good way to determine a suitable penalization value p can be through trial and error. In theory, too large a penalization will lead to worse real Wasserstein metric (6), while a too loose penalization will be not enough to suppress large changes in the design between key frames. For our application, we conducted a parametric study, presented in Figure 7, to demonstrate the effect of using different p values. For each p value, we also calculate the real Wasserstein metric (i.e. using equation (6)) and present the loss (increase) of real Wasserstein metric due to the penalization.

Figure 7 shows that, counter-intuitively, these real Wasserstein metrics do not increase monotonically as the penalization p increases, which is different from our analysis. Our postulation is that such behavior might stem from the nonlinear and non-convex nature of the optimization problem with multiple local minima. Regardless this phenomenon, Figure 7 also shows that the overall designs using $p = 2, 3, 4$ are similar in general, and the corresponding real Wasserstein metrics ($132.958, 119.542$ and 124.292) are within the same order of magnitude. As a result, for our numerical study, we use $p = 2$ in the following results, unless otherwise specified.

Optimization problem (5) is numerically solved using the SNOPT optimizer [27] via a python wrapper [28]. The underlying finite element problem is solved using a python code adapted from the implementation in [29].

Figure 8 shows a series of optimized key frames with a two-level refinement. Figure 8a illustrates the reconfiguration process from an initial structure where the left 40% of the domain is filled with solid and rest left as void. The last frame in Figure 8a also has a volume of 40%. As a result, the volume constraint in problem (5) is effectively requiring all key frames to have a volume that is close to 40% of the entire domain. Figure 8a demonstrates a coarse trend of reconfiguration; in order to gain more details, on-demand refinement can be performed. In this case, we think the reconfiguration between the last design frame and final target in Figure 8 is not clear. Accordingly, we perform two consecutive refinements, as shown in Figure 8b and Figure 8c.

For Figure 8 it can be seen that in the first frames, optimizer moves material to form the thick truss members on the left hand side. Then in later frames, material is stripped from the thick members and used to gradually form the truss members on the right hand side. A consistent trend of moving materials from left to right and gradually forming the truss members can be observed across different level of refinements, and demonstrates the plausibility of the proposed hierarchical approach.

To support our choice of the exponential parameter in the cost constraint (8), we conducted a parametric study to understand this parameter affects the optimization process, as shown in Figure 7. This figure illustrates three optimized series with same initial and target configurations. General consistency can be observed, except for the fourth frame in each series where there is a difference in the local structure in the middle region.

Next, we demonstrate the effect of discretization, and discuss a limitation of the Wasserstein metric that we use as the cost metric.

Figure 9a presents the same series as in Figure 8b. In Figure 9b, we perform another refinement between two design frames to gain a better understanding on the small segment in the middle region. Along with Figure 9b, Figure 9c presents the original continuous density field without post-processing for discrete $0-1$ design. It can be seen in the fifth frame in Figure 9b, we have a tiny piece of floating structure, which in fact is an artifact of the relatively coarse mesh. From the fourth and fifth frames in Figure 9c it can be seen that the thin member along the upper structure and lower right tip is forming gradually, however since the mesh is coarse, such a fragile member cannot be fully captured when we map to a discrete design, producing the disjoint piece.

Finally, a limitation of the Wasserstein metric can be observed by the existence of the same small segment that we have discussed above. Intuitively, creating but later removing this piece can be thought of as a waste, but it does help to reduce the Wasserstein metric between this frame to the next frame, because it's closer to the final member formed in Figure 9b on the right. As a result, this demonstrates that sometimes the Wasserstein metric does not reflect the true cost of moving materials around locally. Nevertheless, it can

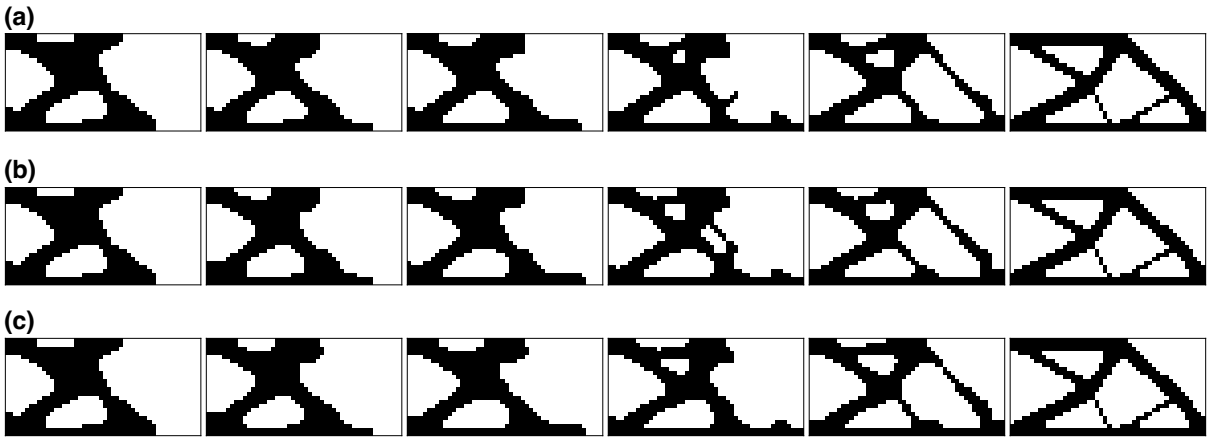


Figure 7: A parametric study for cost constraint. (a) $p = 2, \sum w = 132.958$ $(1.29\times)$; (b) $p = 3, \sum w = 119.542$ $(1.16\times)$; (c) $p = 4, \sum w = 124.292$ $(1.21\times)$. Numbers in parentheses are the ratio between the summed Wasserstein metric between all frames to the Wasserstein metric between the initial and final frames only, i.e. the lower bound. As such, $(1\times)$ would indicate no loss introduced by the penalization.

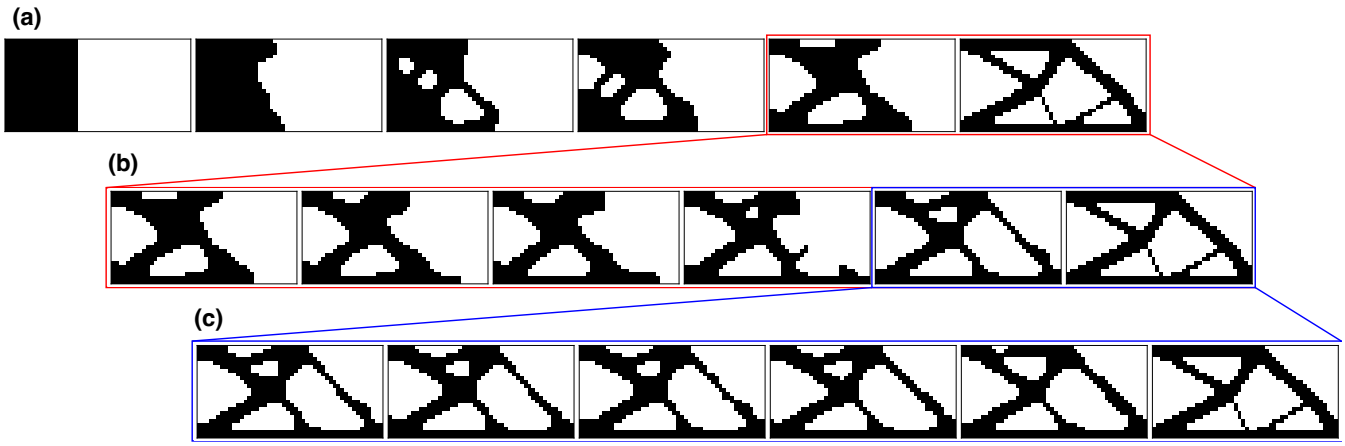


Figure 8: Process design with a two-level hierarchical refinement. (a) Initial optimization from a lumped structure configuration to the target configuration with four key frames, (b) a first-level refinement that adds details between the last design and the target configuration, (c) a second-level refinement between again the last design and the target configuration

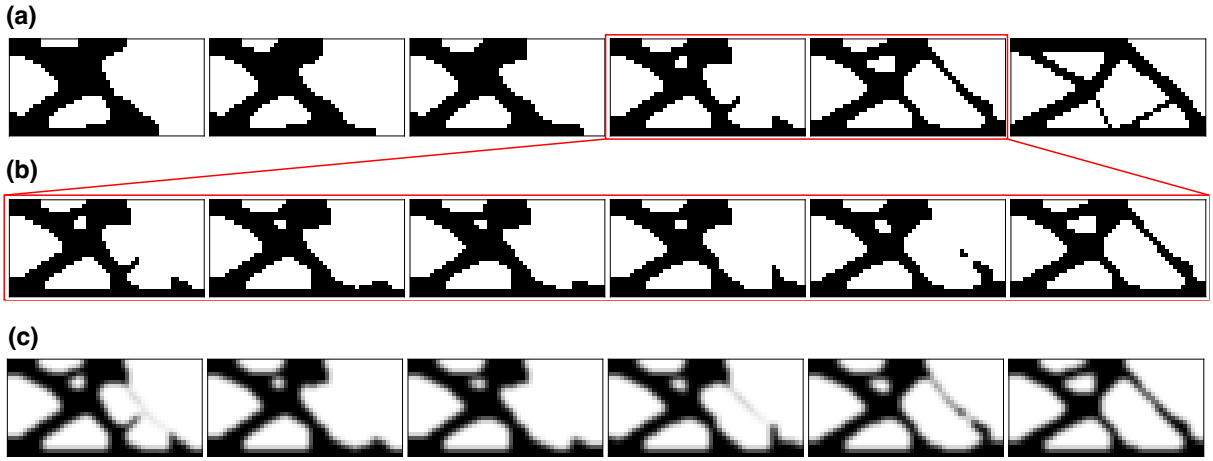


Figure 9: A demonstration of discretization dependency. (a) Design series same as in Figure 8(b). (b) A second-level refinement between the last two design frames in (a). (c) The original topology results without discrete filtering. (b) and (c) demonstrate that the existence of the floating structural artifact in (a) and Figure 8(b) is due to discretization, which is able to be removed or made connected to the main structure via mesh refinement.

still be considered as an effective metric in the global scope.

Robotic Transfer Strategy between Key Frames

To reconfigure between each consecutive pair of key frames x_i and x_{i+1}, we identify the modules that need to be removed, and the modules that need to be added. Denote the set of cells $S = x_i \setminus x_{i+1}$. We shall call S the supply set; it represents the modules that need to be removed from the key frame x_i. Similarly, let $D = x_{i+1} \setminus x_i$ be the set of demand cells; these are the cells that need to be occupied in x_{i+1}. We construct a complete weighted bipartite graph on S and D. For a pair of cells $s \in S$ and $d \in D$ we set the weight of the corresponding edge to be the square of the length of the shortest path from s to d through the interior of the robot structure. Thus we minimize the lengths of all the trajectories while penalizing distant pair matchings. If the sizes of S and D are not equal, we add additional modules at the depot (located along the left border of the configuration) to balance the two sets.

A minimum weight maximum matching in the resulting bipartite graph gives us an assignment of the initial and the target positions of the metamodules to be reconfigured. We connect each pair of the initial and the target positions with a shortest path through the interior of the robot structure (see Figure 10).

The overlay of the resulting paths gives us a directed graph that we use as the reconfiguration network. As each cell of the structure can sustain at most two metamodules at a time, we need to incorporate collision avoidance into our reconfiguration approach. To this extent, we formulate a network flow problem. We add a time component to the reconfiguration network, connecting the cells from the layer corresponding to time t to themselves and the neighboring cells in the layer $t + 1$. As the initial value for the number of time slices, we take the length of the longest path from the minimum weight matching. We set the capacities of the nodes and the edges to be 1, connect an auxiliary source node to all the orange cells in layer $t = 0$, and all the green cells in layer $t = T$ to an auxiliary sink node. The maximum flow between the source and the sink in this network corresponds to a conflict-free reconfiguration schedule. In the case when the value of the maximum flow does not match the number of orange or green cells, we can iteratively increment the value of T, until the resulting flow matches the number of reconfiguring modules. However, in the experiment depicted in Figure 10, this was not necessary.

8. CONCLUSIONS

This work clearly demonstrates the validity of the presented framework for problems of autonomous reconfiguration and assembly. Robotic path planning is well integrated with topology optimization results used to approximately solve an otherwise intractable problem. The framework also allows full consideration to be given to other real world constraints like those of structural integrity and connectedness.

We have also shown here that itinerant robotic assembly can be used to construct complex structures from basic starting configurations. For large extraterrestrial structures, robotic assembly, and consequently proper assembly sequencing methods, will be crucial. This work takes a promising step towards meeting those requirements.

Although this work only demonstrates two-dimensional examples, the proposed framework can be extended to three-dimensional problems, albeit with increased computational cost as the number of variables increase. Similarly, the robotic model can readily be extended into three-dimensions, and, as mentioned throughout the work, the presented framework is malleable enough to accommodate alternative robotic models with minor adjustments.

Future Work

All that being said, there are several ways in which the presented framework can be improved. One such opportunity is with regard to the Wasserstein metric as a metric. The Wasserstein metric does not scale well as the size of the design domain increases, so for larger designs we need to determine or create a better metric for calculating the reconfiguration cost associated with two different structural states. Overall, the Wasserstein metric has presented several issues throughout the development of this work, so even though the Wasserstein metric was a useful metric for measuring assembly cost between frames, a better metric should be found for future development. This goes hand in hand with another direction for future work, which is increasing the coupling between the robotic algorithms and the structural analysis and topology/shape optimization. We believe the results of this work can be considerably improved by incorporating the robotic path planning and other robotic considerations more closely with the structural considerations and optimization process.

Furthermore, even though it is a good approximation in our case study, the Wasserstein metric does not perfectly reflect the distance the robots travel through the structure during the reconfiguration process. Incorporating a more complex metric into the topology optimization step would lead to even more significant computation demands. Thus, coupling the robot path planning phase with the topology optimization, could help us optimize for the same metric in both phases, and thus would lead to better overall performance of the presented architecture.

One of the shortcomings of this work is a lack of control over the structural integrity of intermediate states between the well defined and optimized assembly sequence frames, that is during the robotic reconfiguration. This issue was circumvented in this work by adjusting the optimization constraints to limit the uncertainty of these intermediate states. However, a more general solution would be to impose bounds or constraints on the properties of the interstitial structures as the robotic reconfiguration is occurring. For example, by constraining the properties of the structure formed by the components which remain the same between two frames of the assembly sequence, that is the intersection of two sequential structural designs must also adhere to some metric of structural integrity.

During this work, and on reflection and retrospection, we arrived at several modifications by which the formulation of the topology optimization problem can be improved. In this paper, we use an approximate measure of the total assembly cost, based on the Wasserstein metric, as a constraint on the optimization, but in future work, we will be using the approximation of the total assembly cost as the objective function to be minimized in place of the compliance. The motivation for this change is to improve alignment between the optimization formulation and the desired result. Specifically, we want to minimize the total cost of the assembly, which is why using the approximate total assembly cost as the objective to be minimized is a logical choice. Since, in doing so we replace the compliance objective, which served to impose structural

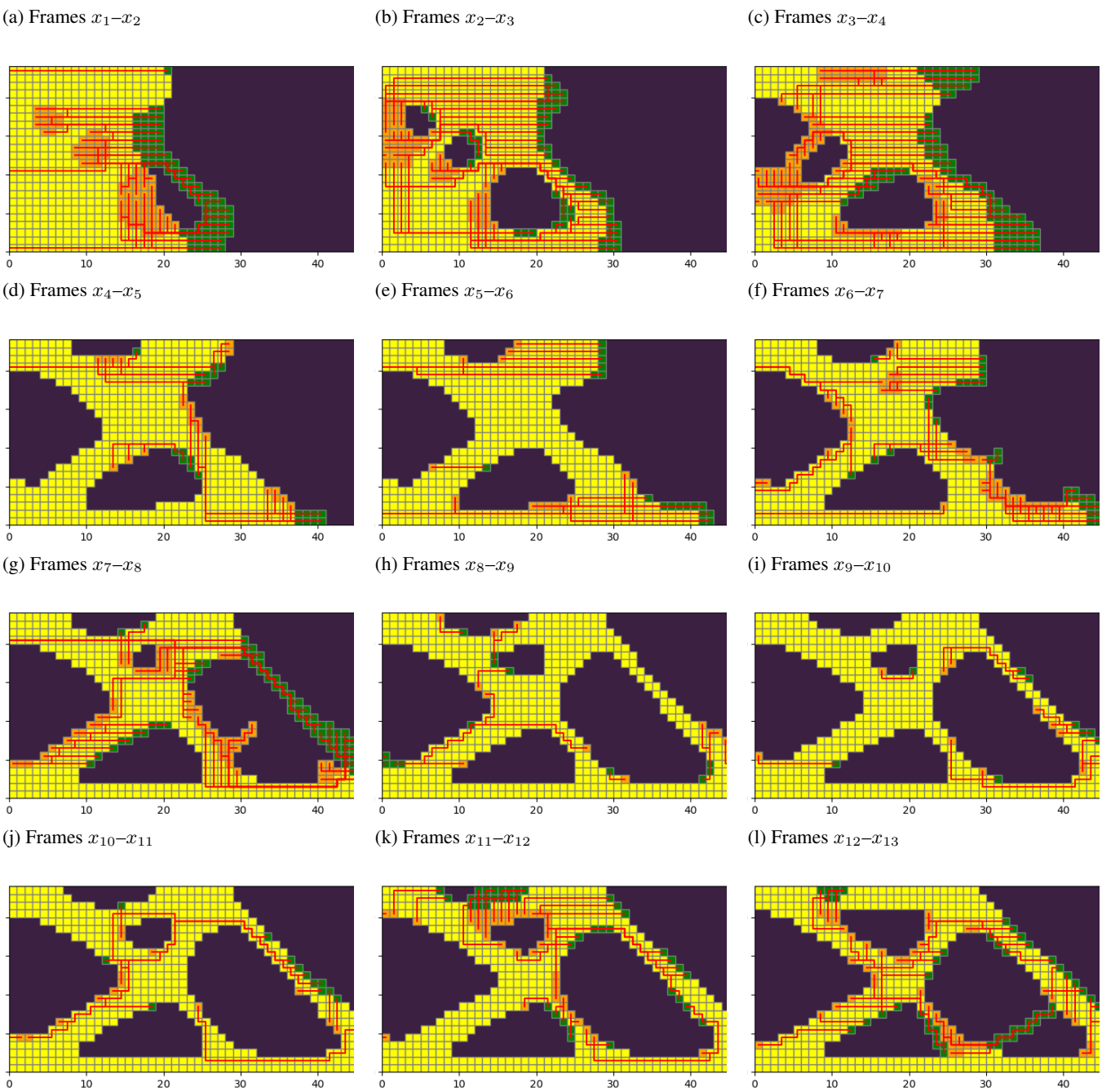


Figure 10: Reconfiguration plans between key frames x_2–x_{13} of the optimization process from Figure 8. Reconfiguration step between keyframes $x_{init} = x_0$ and x_1 is omitted. The cells to be removed in each reconfiguration step are shown in orange, the cells to be added are shown in green, the cells that remain unchanged are shown in yellow. In red are depicted the reconfiguration trajectories that the metamodules will take to move from the orange to the green cells. Additional modules, when needed, are obtained from the depot, located along the left boundary of the structure.

integrity on the intermediate states, a new constraint will be needed to achieve the same goal. One option is to reuse the compliance as a constraint, but a much better option is to implement a constraint on the maximum stress within the structure. This way, we can constrain the maximum stress in the intermediate states to be some factor less than the critical stress of the building blocks, which much more accurately aligns with the desired structural integrity.

ACKNOWLEDGMENTS

This work was supported by the NASA Space Technology Mission Directorate's Game Changing Development Program through the Automated Reconfigurable Mission Adaptive Digital Assembly Systems project.

REFERENCES

[1] A. Ekblaw, E. Shuter, and J. A. Paradiso, *Self-Assembling Space Architecture: tessellated shell structures for space habitats*, ser. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan 2019, 0. [Online]. Available: https://doi.org/10.2514/6.2019-0481

[2] H. H. Chaudhary, "Unify: Self-assembling shell structures for space exploration," *International Journal of Research in Science and Technology*, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:219123357

[3] J. Rouvinet, A. Ummel, F. Cosandier, D. Nguyen, and V. Schaffter, "PULSAR: development of a mirror tile prototype for future large telescopes robotically assembled in space," in *Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation IV*, R. Navarro and R. Geyl, Eds., vol. 11451, International Society for Optics and Photonics. SPIE, 2020, p. 1145142. [Online]. Available: https://doi.org/10.1117/12.2576237

[4] D. Li, L. Zhong, W. Zhu, Z. Xu, Q. Tang, and W. Zhan, "A survey of space robotic technologies for on-orbit assembly," *Space: Science & Technology*, vol. 2022, 2022. [Online]. Available: https://spj.science.org/doi/abs/10.34133/2022/9849170

[5] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.1254295

[6] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sambot: A self-assembly modular robot for swarm robot," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 66–71.

[7] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1115–1130, 2006.

[8] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197 – 224, 1988.

[9] M. Cavazzuti, A. Baldini, E. Bertocchi, D. Costi, E. Torricelli, and P. Moruzzi, "High performance automotive chassis design: a topology optimization based approach," *Structural and Multidisciplinary Optimization*, vol. 44, no. 1, pp. 45–56, Jul 2011. [Online]. Available: https://doi.org/10.1007/s00158-010-0578-7

[10] N. Stoiber and B. Kromoser, "Topology optimization in concrete construction: a systematic review on numerical and experimental investigations," *Structural and Multidisciplinary Optimization*, vol. 64, no. 4, pp. 1725–1749, Oct 2021. [Online]. Available: https://doi.org/10.1007/s00158-021-03019-6

[11] J.-H. Zhu, W.-H. Zhang, and L. Xia, "Topology optimization in aircraft and aerospace structures design," *Archives of Computational Methods in Engineering*, vol. 23, no. 4, pp. 595–622, Dec 2016. [Online]. Available: https://doi.org/10.1007/s11831-015-9151-2

[12] G. K. Das, J. T. Gloyd, Y. Fu, G. Kennedy, and K. James, *Topology optimization of crash-tolerant aircraft structures*, ser. AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Jun 2023, 0. [Online]. Available: https://doi.org/10.2514/6.2023-4379

[13] Y. Fu, C. Smith, B. Li, and G. Kennedy, *Simultaneous Trajectory and Topology Optimization of Flexible Multibody Systems*, ser. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan 2024, 0. [Online]. Available: https://doi.org/10.2514/6.2024-2232

[14] D. Herrero-Pérez and P. J. Martínez Castejón, "Multi-gpu acceleration of large-scale density-based topology optimization," *Advances in Engineering Software*, vol. 157-158, p. 103006, 2021. [Online]. Available: https://doi.org/10.1016/j.advengsoft.2021.103006

[15] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, "Giga-voxel computational morphogenesis for structural design," *Nature*, vol. 550, no. 7674, pp. 84–86, Oct 2017. [Online]. Available: https://doi.org/10.1038/nature23911

[16] Y. Fu and G. J. Kennedy, "Quasi-newton corrections for compliance and natural frequency topology optimization problems," *Structural and Multidisciplinary Optimization*, vol. 66, no. 8, p. 176, Jul 2023. [Online]. Available: https://doi.org/10.1007/s00158-023-03630-9

[17] Z. Kang, J. He, L. Shi, and Z. Miao, "A method using successive iteration of analysis and design for large-scale topology optimization considering eigenfrequencies," *Computer Methods in Applied Mechanics and Engineering*, vol. 362, p. 112847, 2020. [Online]. Available: https://doi.org/10.1016/j.cma.2020.112847

[18] B. Li, Y. Fu, and G. J. Kennedy, "Topology optimization using an eigenvector aggregate," *Structural and Multidisciplinary Optimization*, vol. 66, no. 10, p. 221, Oct 2023. [Online]. Available: https://doi.org/10.1007/s00158-023-03674-x

[19] H. Li, S. Knapik, Y. Li, C. Park, J. Guo, S. Mojumder, Y. Lu, W. Chen, D. W. Apley, and W. K. Liu, "Convolution hierarchical deep-learning neural network tensor decomposition (c-hidenn-td) for high-resolution topology optimization," *Computational Mechanics*, vol. 72, no. 2, pp. 363–382, Aug 2023. [Online]. Available: https://doi.org/10.1007/s00466-023-02333-8

[20] D. Rus and M. Vona, "Crystalline robots: Self-reconfiguration with compressible unit modules," *Auton. Robots*, vol. 10, no. 1, pp. 107–124, 2001. [Online]. Available: https://doi.org/10.1023/A:1026504804984

[21] G. Aloupis, S. Collette, M. Damian, E. D. Demaine, R. Y. Flatland, S. Langerman, J. O'Rourke, V. Pinciu, S. Ramaswami, V. Sacristán, and S. Wuhrer, "Efficient constant-velocity reconfiguration of crystalline robots," *Robotica*, vol. 29, no. 1, pp. 59–71, 2011. [Online]. Available: https://doi.org/10.1017/S026357471000072X

[22] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-451.html

[23] O. Pele and M. Werman, "A linear time histogram metric for improved SIFT matching," in *Computer Vision – ECCV 2008*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer, 2008, pp. 495–508.

[24] G. Kennedy and Y. Fu, "Topology optimization benchmark problems for assessing the performance of optimization algorithms," in *AIAA Scitech 2021 Forum*, ser. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan 2021, 0. [Online]. Available: https://doi.org/10.2514/6.2021-1357

[25] M. P. Bendsøe and O. Sigmund, *Topology optimization: theory, methods and applications*. Springer, 2003.

[26] M. Stolpe and K. Svanberg, "An alternative interpolation scheme for minimum compliance topology optimization," *Structural and Multidisciplinary Optimization*, vol. 22, pp. 116–124, 2001.

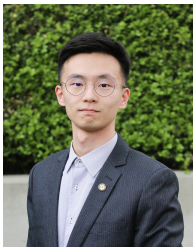
[27] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *SIAM Review*, vol. 47, no. 1, pp. pp. 99–131, 2005.

[28] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins, "pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems," *Journal of Open Source Software*, vol. 5, no. 54, p. 2564, 2020.

[29] B. Li, Y. Fu, and G. J. Kennedy, "Topology optimization using an eigenvector aggregate," *Structural and Multidisciplinary Optimization*, vol. 66, no. 10, p. 221, Oct 2023. [Online]. Available: https://doi.org/10.1007/s00158-023-03674-x

BIOGRAPHY

James Todd Gloyd *received his B.S., M.S., and Ph.D degrees in aerospace engineering from the Georgia Institute of Technology, where he was a research fellow under the NASA Space Technology Research Grant. His work is currently in structural analysis and topology optimization for metamaterials and discrete lattice structures, and he has worked closely in tensegrity structures research and analysis of pressure vessels. He is currently a member of the Coded Structures Laboratory at NASA Ames, and previously a member of the Structures and Multidisciplinary Design Optimization Research Group under the direction of Dr. Graeme Kennedy and a member of the Computational Solid Mechanics Laboratory under the direction of Dr. Julian Rimoli.*



Yicong Fu *is a Ph.D. candidate in aerospace engineering at the Structures and Multidisciplinary Design Optimization Group supervised by Dr. Graeme Kennedy at the Georgia Institute of Technology, where he received his M.S. in aerospace engineering and M.S. in computational science and engineering. His research interests include multiphysics analysis and gradient-based optimization, topology optimization, and high performance computing. He received his bachelor's degree in engineering mechanics from Beihang University, China.*

Irina Kostitsyna *is a computer scientist at KBR at NASA Ames Research Center, working on algorithmic foundations for autonomy and multi-robot assembly. Her interests include computational geometry, distributed geometric algorithms, and planning for multi-agent systems. She received her Ph.D. in computer science from Stony Brook University.*